EPCIO Series Motion Control Command Library Integrated Testing Environment User Manual

(Applicable to Motion Control Command Library V.5.10)

Version: **V.5.10**

Date: **2009.10**

http://www.epcio.com.tw

Table of Contents

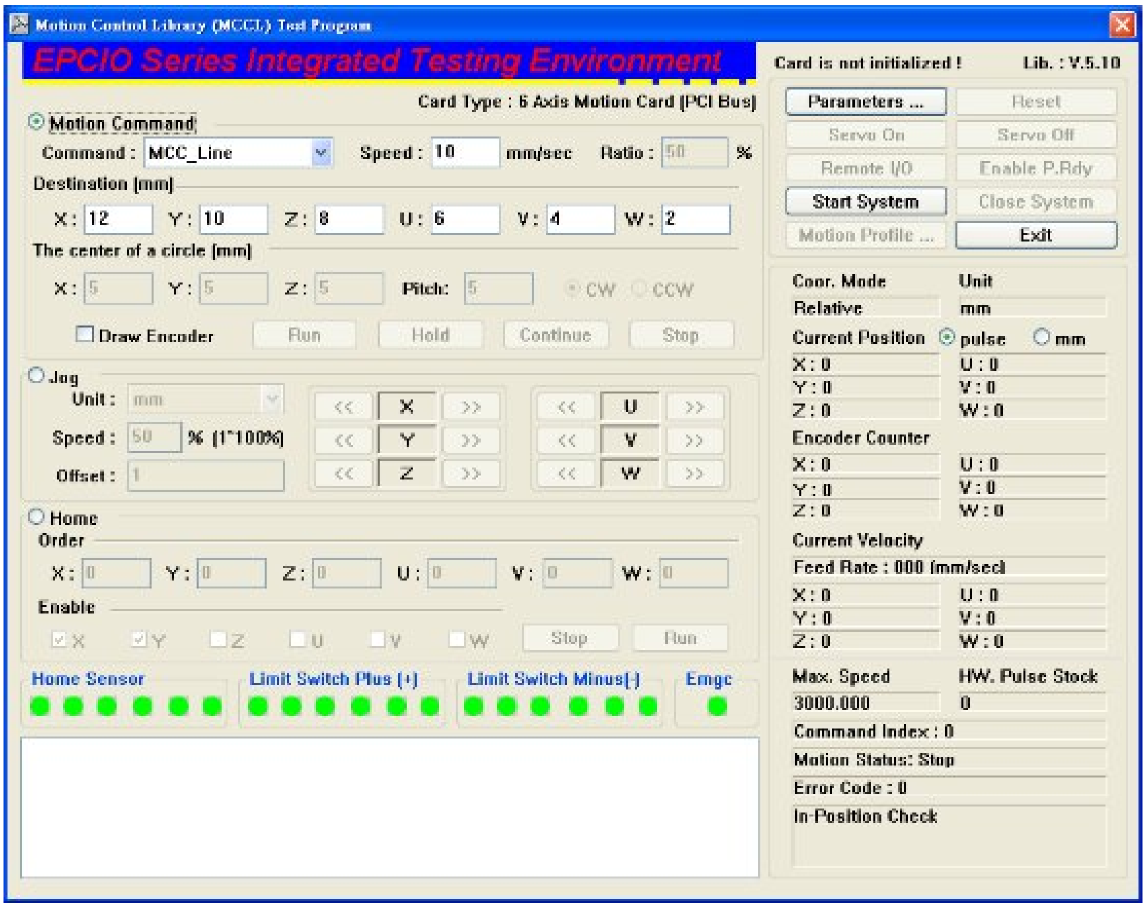
I. Introduction to the Motion Control Command Library (MCCL) Testing Software

Motion control command library testing software is used to test single EPCIO Series motion control card using a single group (for a description of groups, please refer to the "**EPCIO Series Motion Control Command Library User Manual"**). The group's parameter settings are shown below:

```
m_nGroupIndex = MCC_CreateGroup(
                    0, // X-axis programming results from output Channel 0
                    1, // Y-axis programming results from output Channel 1
                    2, // Z-axis programming results from output Channel 2
                    3, // U-axis programming results from output Channel 3
                    4, // V-axis programming results from output Channel 4
                    (four-axes card is -1)
                    5, // W-axis programming results from output Channel 5
                    (four-axes card is -1)
                    0); // Control card number for this group's response
```

Therefore, if the group number needs to be input into the commands used in the software, m_nGroupIndex can be specified without exception.

Testing software can be used only in terms of the basic function commands provided by MCCL. Regarding methods for other commands, please refer to the description in the "**EPCIO Series Motion Control Command Library Reference Manual**." For questions concerning command uses, please consult the "**EPCIO Series Motion Control Command Library Examples Manual**." Next, pictures will be used to illustrate basic function commands and clearly explain the operational methods for the testing software. Fig. 1 is the main screen for the motion control command library testing software.


Motion Control Library (MCCL) Test Program

EPCIO Series Integrated Testing Environment

Card is not initialized !   Lib. : V.5.10

Card Type : 6 Axis Motion Card (PCI Bus)

[✓ Motion Command]

Command : MCC_Line   Speed : 10 mm/sec   Ratio : 50 %

Destination (mm)

X : 12   Y : 10   Z : 8   U : 6   V : 4   W : 2

The center of a circle (mm)

X : 5   Y : 5   Z : 5   Pitch : 5   [✓ CW] CCW

Draw Encoder

Jog

Unit : mm   Speed : 50 % (1~100%)   Offset : 1

Home

Order

X : 0   Y : 0   Z : 0   U : 0   V : 0   W : 0

Enable

[✓ X] [✓ Y] Z U V W

Coor. Mode: Relative   Unit: mm

Current Position [✓ pulse] mm

X : 0   U : 0
Y : 0   V : 0
Z : 0   W : 0

Encoder Counter

X : 0   U : 0
Y : 0   V : 0
Z : 0   W : 0

Current Velocity

Feed Rate : 000 (mm/sec)

X : 0   U : 0
Y : 0   V : 0
Z : 0   W : 0

Max. Speed: 3000.000   HW. Pulse Stock: 0

Command Index : 0

Motion Status: Stop

Error Code : 0

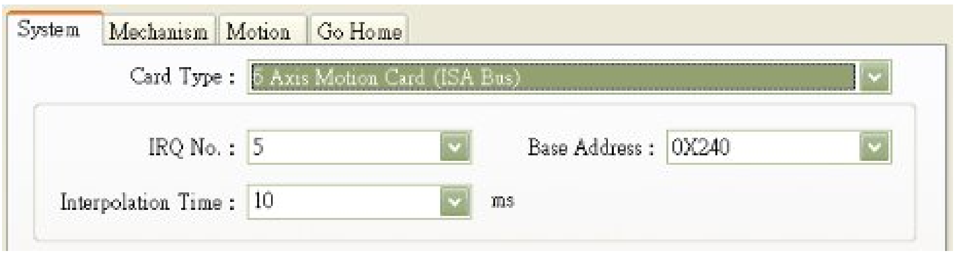
In-Position Check

Fig. 1

II. Testing Software Activation

To test the MCCL functions, the MCCL must first be activated using the following procedure:

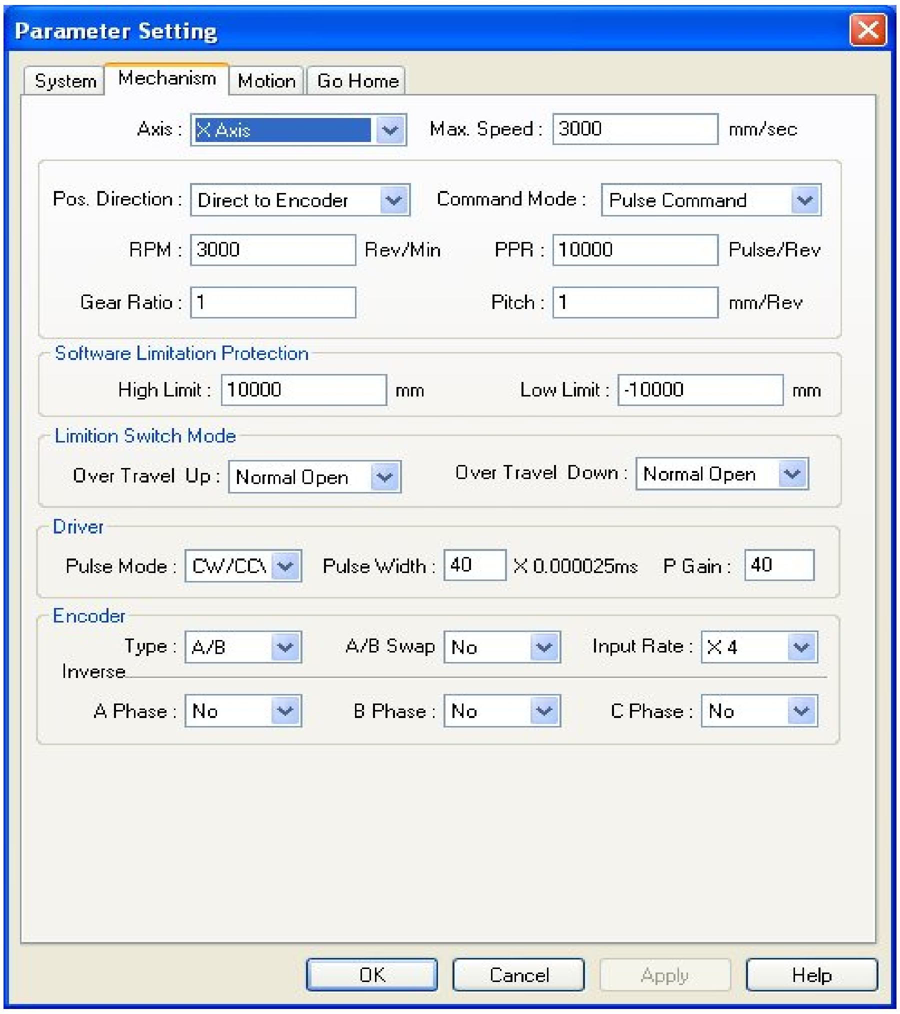
1. The accuracy of the set values for the card type, base address, and IRQ number (setting the base address and IRQ number is not required when using the EPCIO Series PCI Motion Control Card) can each be inspected on the **"System Parameters Settings Page"** (see Fig. 2). Furthermore, the interpolation time (with a suggested value of 5 ms) can also be set here.



Card Type : 6 Axis Motion Card (ISA Bus)
IRQ No. : 5
Base Address : 0X240
Interpolation Time : 10 ms

Fig. 2

2. The accuracy of the set Mechanism parameters can be inspected for each setting on the “**Mechanism Parameters Settings Page**” (see Fig. 3). For the meaning of each parameter in Fig. 3, please refer to “**EPCIO Series Motion Control Command Library User Manual**.”


Parameter Setting

System | Mechanism | Motion | Go Home

Axis : X Axis   Max. Speed : 3000 mm/sec

Pos. Direction : Direct to Encoder   Command Mode : Pulse Command
RPM : 3000 Rev/Min   PPR : 10000 Pulse/Rev
Gear Ratio : 1   Pitch : 1 mm/Rev

Software Limitation Protection
High Limit : 10000 mm   Low Limit : -10000 mm

Limition Switch Mode
Over Travel Up : Normal Open   Over Travel Down : Normal Open

Driver
Pulse Mode : CW/CCW   Pulse Width : 40 X 0.000025ms   P Gain : 40

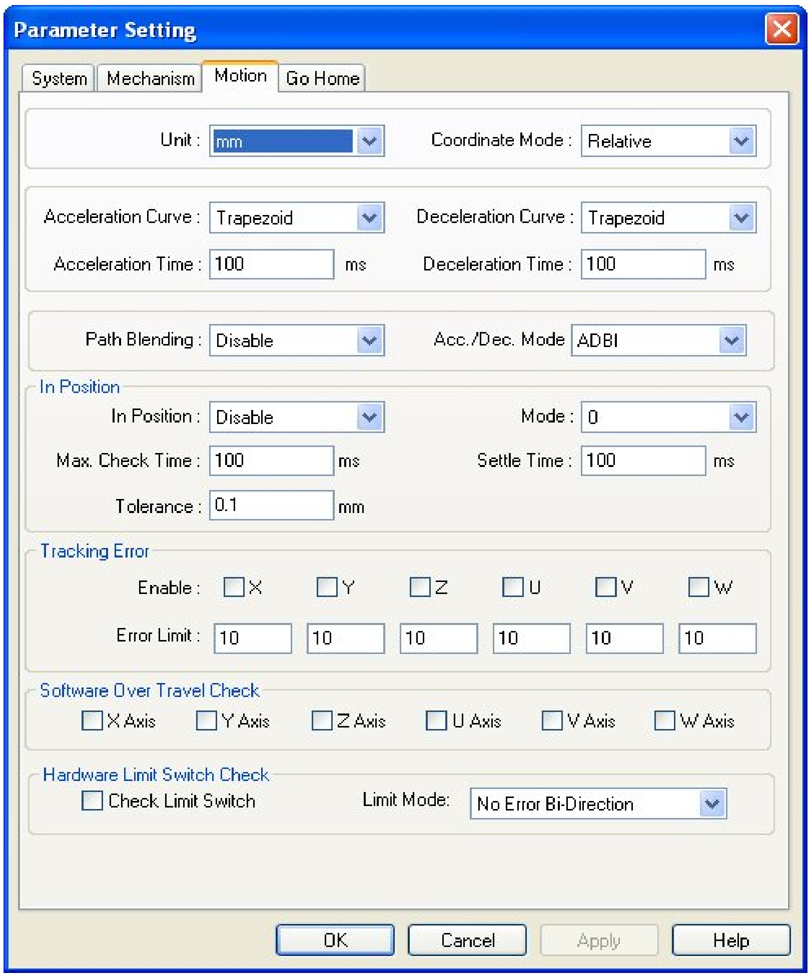
Encoder
Type : A/B   A/B Swap : No   Input Rate : X 4
Inverse
A Phase : No   B Phase : No   C Phase : No

Fig. 3

3. Clicking the Start System button will not only set the system parameters, but will also call MCC_InitSystem().

III. Motion Property Settings

This section describes the necessary MCCL commands used in the testing software when changing the settings in the "**Motion Property Settings Page,**" and how the user can learn methods for using MCCL according to changes in motion trajectory. Fig. 4 is the "**Motion Property Settings Page**."



Parameter Setting

System | Mechanism | Motion | Go Home

Unit : mm
Coordinate Mode : Relative

Acceleration Curve : Trapezoid
Deceleration Curve : Trapezoid
Acceleration Time : 100 ms
Deceleration Time : 100 ms

Path Blending : Disable
Acc./Dec. Mode ADBI

In Position
In Position : Disable
Mode : 0
Max. Check Time : 100 ms
Settle Time : 100 ms
Tolerance : 0.1 mm

Tracking Error
Enable : ☐ X ☐ Y ☐ Z ☐ U ☐ V ☐ W
Error Limit : 10 10 10 10 10 10

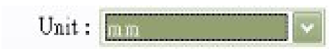
Software Over Travel Check
☐ X Axis ☐ Y Axis ☐ Z Axis ☐ U Axis ☐ V Axis ☐ W Axis

Hardware Limit Switch Check
☐ Check Limit Switch
Limit Mode: No Error Bi-Direction

OK | Cancel | Apply | Help

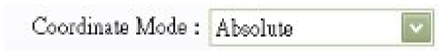
Fig. 4

The command calls corresponding to each selection item are described below.

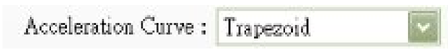
Unit 

Unit : mm

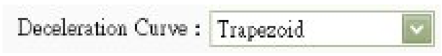
Sets the unit used in displacement. When selecting "mm," it calls MCC_SetUnit (UNIT_MM). When selecting "inch," it calls MCC_SetUnit (UNIT_INCH).

Coordinate Mode 
Coordinate Mode : Absolute

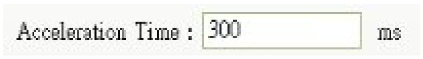
Sets whether each axial coordinate position is expressed in the absolute mode or the incremental mode. When selecting “Absolute,” it calls MCC_SetAbsolute(). When selecting “Relative,” it calls MCC_SetIncrease().

Acceleration Curve 
Acceleration Curve : Trapezoid

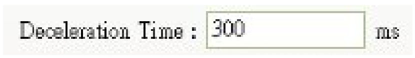
Sets the acceleration type for axes X, Y, Z, U, V, and W as either a trapezoidal or an S curve for line, curve, or circular movements. When selecting “Trapezoid,” it calls MCC_SetAccType('T'), indicating the use of a trapezoidal acceleration curve. When selecting “S,” it calls MCC_SetAccType('S'), indicating the use of an S-shaped acceleration curve.

Deceleration Curve 
Deceleration Curve : Trapezoid

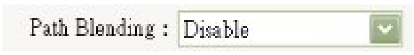
Sets the deceleration type for axes X, Y, Z, U, V, and W as either a trapezoidal or an S curve for line, curve, or circular movements. When selecting “Trapezoid,” it calls MCC_SetDecType('T'), indicating the use of a trapezoidal deceleration curve. When selecting “S,” it calls MCC_SetDecType('S'), indicating the use of an S-shaped deceleration curve.

Acceleration Time 
Acceleration Time : 300 ms

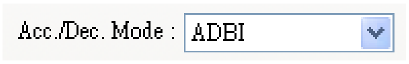
Sets the acceleration time in units of ms. The set acceleration time must be greater than 0. Assuming that the acceleration time dfTime is required, it simply calls MCC_SetAccTime (dfTime).

Deceleration Time 
Deceleration Time : 300 ms

Sets the deceleration time in units of ms. The set deceleration time must be greater than 0. Assuming that the deceleration time dfTime is required, it simply calls MCC_SetAccTime (dfTime).

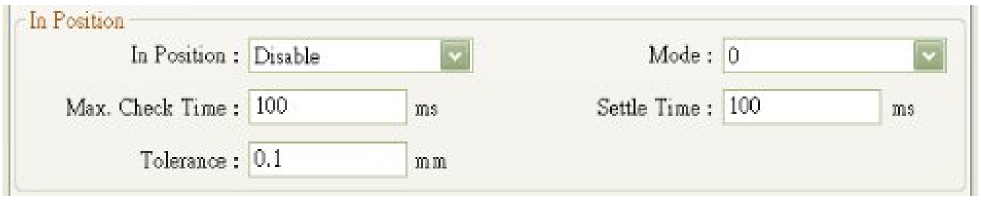
Path Blending 
Path Blending : Disable

Enables path blending. Selecting "Disable" disables path blending by calling MCC_DisableBlend(). Selecting "Enable" enables path blending by calling MCC_EnableBlend().

Acc./Dec. Mode 
Acc./Dec. Mode : ADBI

Sets the acceleration/deceleration mode for axes X, Y, Z, U, V, and W in line, curve, or circular movements as either acceleration/deceleration before interpolation, or acceleration/deceleration after interpolation. When selecting "ADBI," it calls MCC_SetAccDecMode('B'), indicating the use of acceleration/deceleration before interpolation mode. When selecting "ADAI," it calls MCC_SetAccDecMode('ADAI'), indicating the use of acceleration/deceleration after interpolation mode.

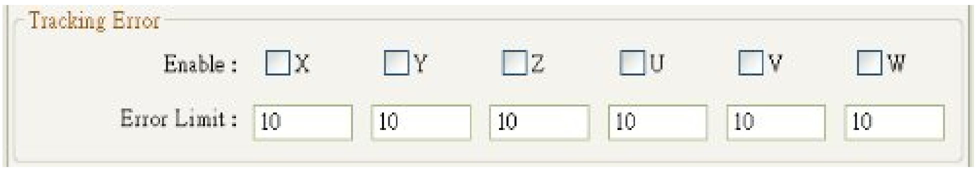
In Position


In Position
In Position : Disable
Mode : 0
Max. Check Time : 100 ms
Settle Time : 100 ms
Tolerance : 0.1 mm

Enables the in-position confirmation function and sets the related parameters (for information regarding the in-position confirmation function, please refer to the **EPCIO Series Motion Control Command Library User Manual**).

In Position : Disable	MCC_EnableInPos / MCC_DisableInPos
Mode : 0	MCC_SetInPosMode
Max. Check Time : 100 ms	MCC_SetInPosMaxCheckTime
Settle Time : 100 ms	MCC_SetInPosSettleTime
Tolerance : 0.1 mm	MCC_SetInPosToleranceEx

Tracking Error



Enable :	☐ X	☐ Y	☐ Z	☐ U	☐ V	☐ W
Error Limit :	10	10	10	10	10	10

Enables the tracking error function and sets the related parameters (for information regarding the tracking error function, please refer to the **EPCIO Series Motion Control Command Library User Manual**).

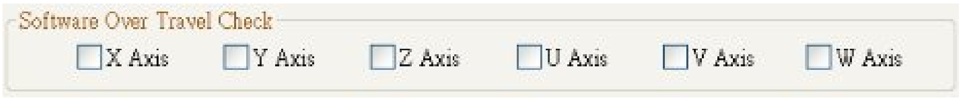
Enable/Disable Tracking Error Function:

MCC_EnableTrackError/MCC_DisableTrackError

Set Tracking Error Permissible Limits:

MCC_SetTrackErrorLimit

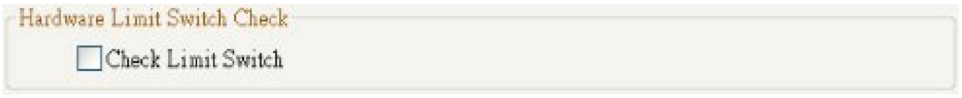
Software Over Travel Check



☐ X Axis ☐ Y Axis ☐ Z Axis ☐ U Axis ☐ V Axis ☐ W Axis

"**Software Over Travel Check**" uses MCC_SetOverTravelCheck to enable the software over travel check function for each axis, limiting the displacement within the work zone.

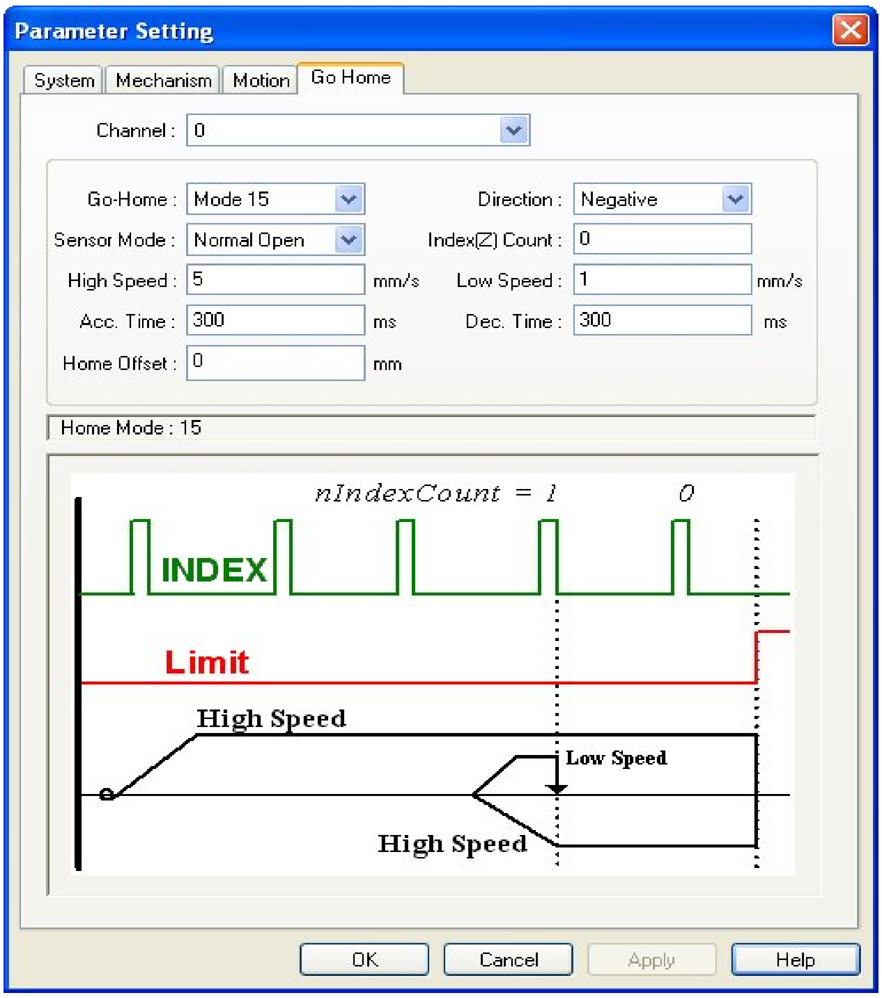
Hardware Limit Switch Check



☐ Check Limit Switch

If ✓ Check Limit Switch is selected, it calls MCC_EnableLimitSwitchCheck to enable the check limit switch function; otherwise, it calls MCC_DisableLimitSwitchCheck to disable the check limit switch function. The user may also call MCC_GetLimitSwitchStatus to check whether the limit switch has currently been touched. Using these commands requires accurately setting the mechanism parameters *wOverTravelUpSensorMode* and *wOverTravelDownSensorMode* (to Normal Open or Normal Close).

IV. Go Home Motion Property Settings



Channel : 0

Go-Home : Mode 15
Direction : Negative
Sensor Mode : Normal Open
Index(Z) Count : 0
High Speed : 5 mm/s
Low Speed : 1 mm/s
Acc. Time : 300 ms
Dec. Time : 300 ms
Home Offset : 0 mm

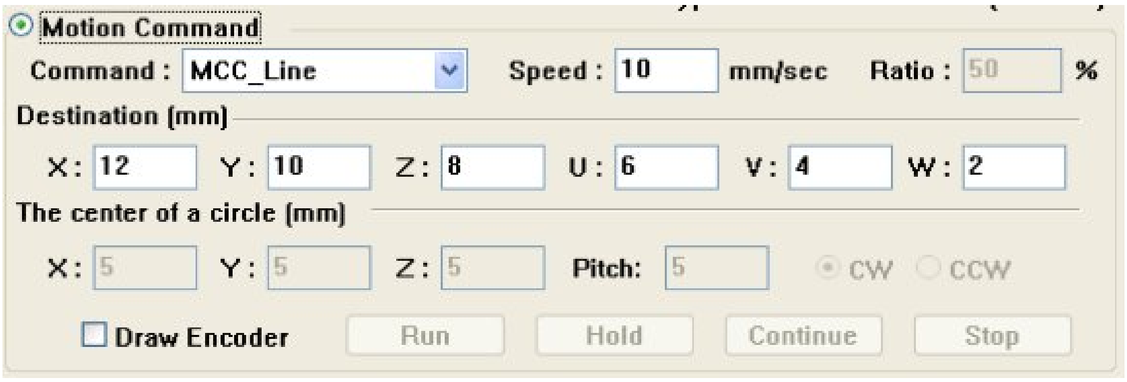
Home Mode : 15

Fig. 5

Fig. 5 shows the property settings related to the Go Home motion. These setting values directly reflect the Go Home parameters. For details, refer to the **“EPCIO Series Motion Control Command Library User Manual**.”

V. Motion Command Execution

Fig. 6 below is the “Motion Command Parameters Settings” section. Content related to general motion operations is described below.



● Motion Command

Command : MCC_Line Speed : 10 mm/sec Ratio : 50 %

Destination (mm)

X : 12 Y : 10 Z : 8 U : 6 V : 4 W : 2

The center of a circle (mm)

X : 5 Y : 5 Z : 5 Pitch: 5 [✓ CW] CCW

☐ Draw Encoder

Fig. 6

Motion Command Selection:

Command : MCC_Line

The motion command type can be selected here. The selection content is identical to the command name, including point-to-point, line, circle, curve, and helix motions.

Speed Settings:

Speed : 10 mm/sec

is used to set the feed speed in units of mm/sec or inch/sec. The input value acts as the call parameter for MCC_SetFeedSpeed, but the value cannot be less than or equal to 0.

Ratio : 50 %

sets the point-to-point speed ratio. The input value can range between 1 and 100, acting as the call parameter for MCC_SetPtPSpeed.

Parameters:

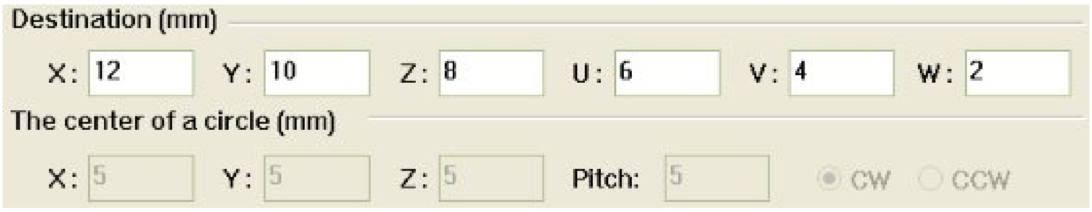
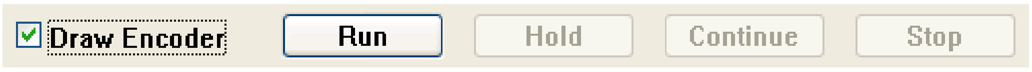


Fig. 7

"Destination" and "**The center of a circle (mm)**" in Fig. 7 are the required parameters for calling the commands listed above. For details, please refer to the **"EPCIO Series Motion Control Command Library User Manual**."

Draw Motor Encoder Feedback Path:

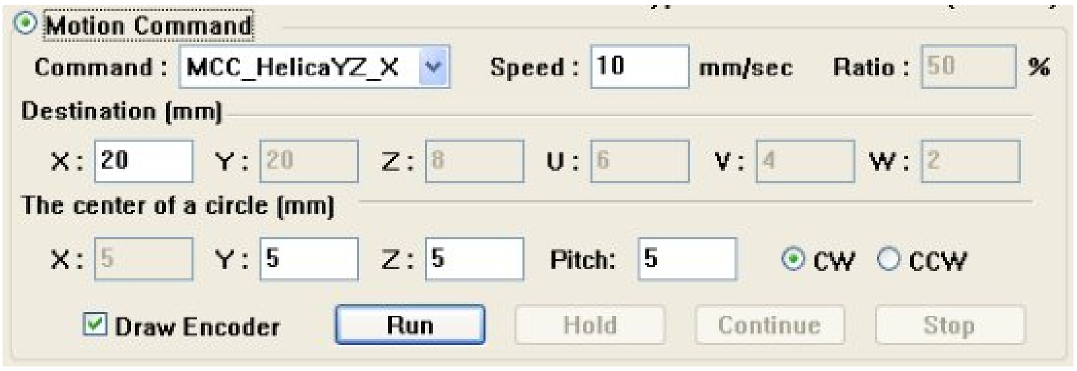


After each parameter is set without error, click the Run button once to send the motion command to the motion command queue. Clicking the Hold button calls MCC_HoldMotion, temporarily pausing motion. Clicking the Continue button calls MCC_ContiMotion, resuming the paused motion command. Clicking the Stop button calls MCC_AbortMotionEx, aborting the current motion and removing the inventory commands in the motion command queue.

To draw the motor encoder feedback path, please select Draw Encoder, and click the Run button again. After the command execution is complete, a new window will pop up drawing the actual motion feedback path of the motor encoder for each axis. As a path verification tool, the red lines are the encoder paths for axes XYZ, and the blue lines are the encoder paths for axes UVW.

The mouse and keyboard can be used in the encoder path window to control the path screen. Left click on the mouse and move it across the screen to rotate the path screen as desired, and use the scroll wheel on the mouse to enlarge or reduce the screen. The directional arrows on the keyboard can be used to translate the entire path screen up, down, left, or right; F3 and F4 can rotate the entire path clockwisely or counter-clockwisely along the X axis; and F5 and F6 can rotate the entire path clockwisely or counter-clockwisely along the Y axis. Once verification is complete, simply close the window.

An example is provided for executing the motion command seen in Fig. 8.



[✓ Motion Command]

Command : MCC_HelicaYZ_X Speed : 10 mm/sec Ratio : 50 %

Destination (mm)

X : 20 Y : 20 Z : 8 U : 6 V : 4 W : 2

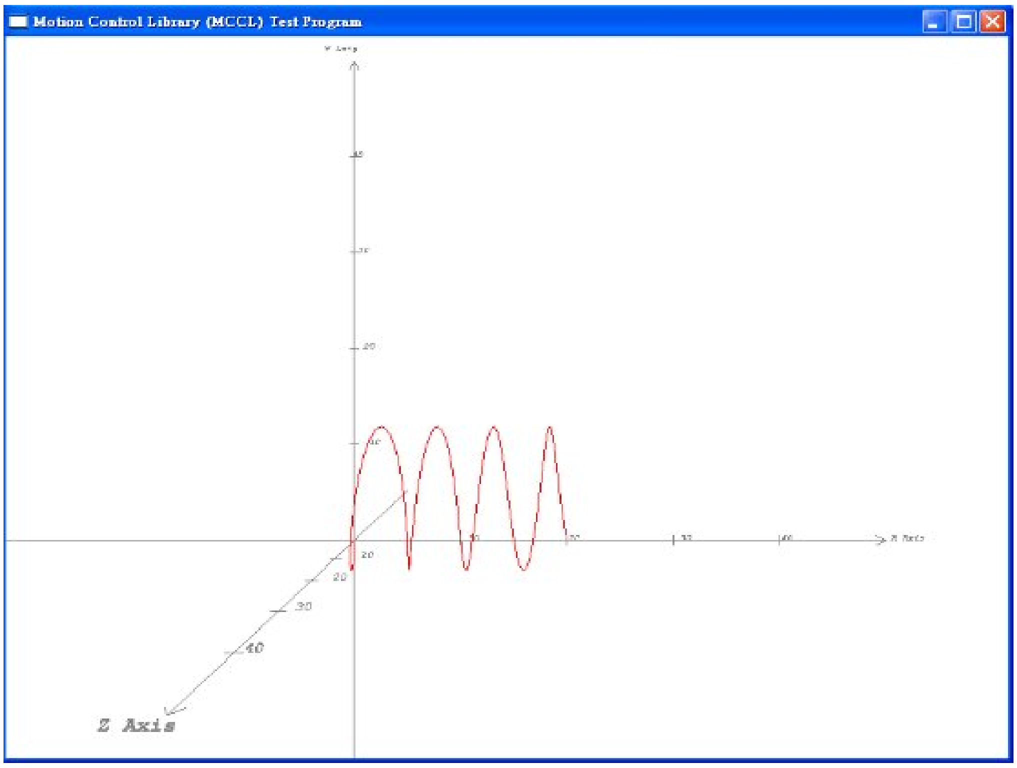
The center of a circle (mm)

X : 5 Y : 5 Z : 5 Pitch: 5 [✓ CW] CCW

[✓ Draw Encoder]

Fig. 8

Click the Run button once. Once the command is executed, the path screen below will pop up.



VI. Jog Motion

Fig. 9 is the "**Jog Motion Parameter Settings**" section. Selections related to the Jog motion are described below.

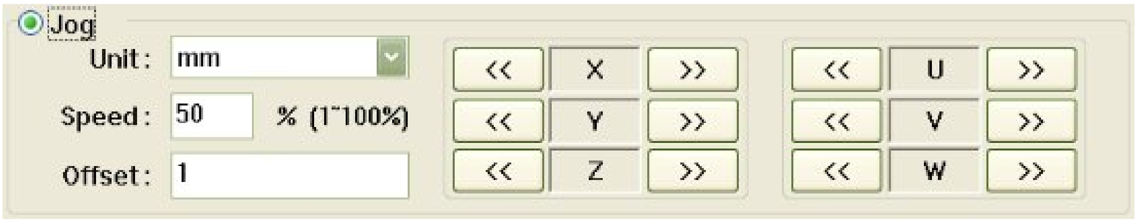


Fig. 9

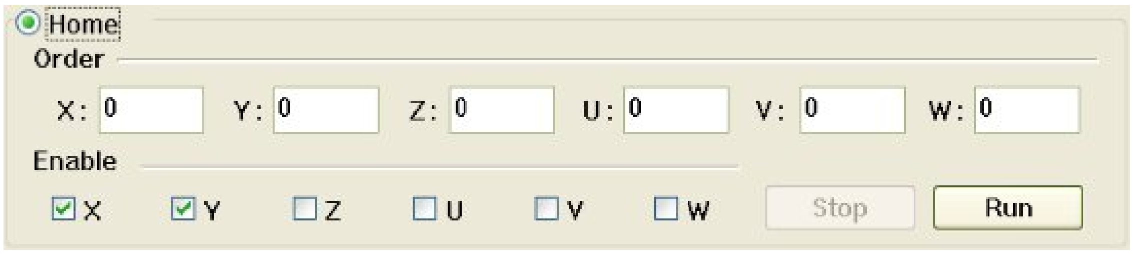
Displacement Unit: Unit: mm

If "**mm**" is selected as the displacement unit, the Jog motion will use mm as the displacement unit. When a Jog motion control button is used (for example >>), the specified axis will be driven according to the appointed incremental displacement value (the input value of Offset: 1) and the feed speed (feed speed is the input value of Speed: 50 % (1~100%) multiplied by the RPM × Pitch/Gear Ratio of each axis). The call here is MCC_JogSpace().

If "Pulse" is selected as the displacement unit, the Jog motion will use pulse as the displacement unit. If the system motion status is stop and the Jog motion control button is used, the specified axis will be driven according to the appointed pulse displacement and direction. The pulse displacement should not be set to an excessively large volume (it cannot exceed 2048 pulses). The call here is MCC_JogPulse().

VII. Go Home Motion

Fig. 10 is the “**Go Home Parameters Settings**” section. Items related to Home motion operations are described below.



X: 0 Y: 0 Z: 0 U: 0 V: 0 W: 0

[✓ X] [✓ Y] ☐ Z ☐ U ☐ V ☐ W

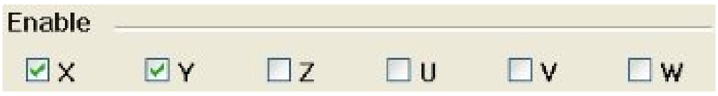
Fig. 10

Commands regarding the Go Home motion are presented below:

```
MCC_Home(    int  nXOrder, int  nYOrder, int  nZOrder,
             int  nUOrder, int  nVOrder, int  nWOrder, WORD wCardIndex);

MCC_GetGoHomeStatus();
```

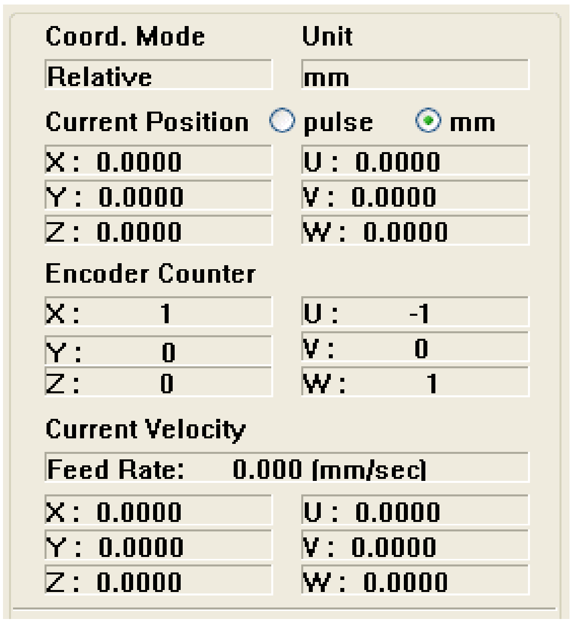
MCC_Home() allows the machine to complete its return home. The command MCC_GetGoHomeStatus() can be coupled with MCC_Home() to check whether the return home is complete. nXOrder to nWOrder is the reset sequence for axes X, Y, Z, U, V, and W, respectively; the reset sequence set value ranges from 0 to 5. These parameters can be obtained in the “**Go Home Parameter Settings**” section.

The feed speed unit for each axis is either mm/sec or inch/sec. The reset sequence for motion axes that have not executed Go Home needs to be set at 0xff(255). In this picture  [✓ X] [✓ Y] ☐ Z ☐ U ☐ V ☐ W, the unselected motion axes have a reset sequence set at 0xff.

After each parameter has been set without error, click the “**Run”** button to call MCC_Home, executing the Go Home motion. During execution, click the “**Stop**” button to call MCC_AbortGoHome, stopping the Go Home motion.

VIII. Motion Status and Information Display

Fig. 11 is the "**Motion Status and Information Display**" section. Methods for obtaining information on each item are described below.



Coord. Mode: Relative
Unit: mm

Current Position ○ pulse [✓ mm]

X : 0.0000	U : 0.0000
Y : 0.0000	V : 0.0000
Z : 0.0000	W : 0.0000

Encoder Counter

X : 1	U : -1
Y : 0	V : 0
Z : 0	W : 1

Current Velocity

Feed Rate: 0.000 (mm/sec)

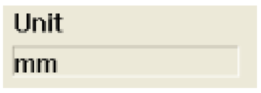
X : 0.0000	U : 0.0000
Y : 0.0000	V : 0.0000
Z : 0.0000	W : 0.0000

Fig. 11

Coordinate Mode:

Coord. Mode: Relative

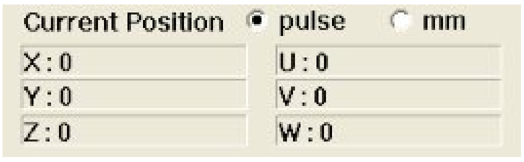
MCC_GetCoordType can be used to acquire the mode currently being used to express coordinate values. A returned value of 0 for this command indicates that the incremental mode is currently being used. A returned value of 1 indicates that the absolute mode is currently being used.

Displacement Units Used: 

Unit: mm

MCC_GetUnit() can be used to obtain the displacement units currently being used. A returned value of UNIT_MM for this command indicates that the metric system (mm) is currently being used. A returned value of UNIT_INCH indicates that the imperial system (inch) is currently being used.

Cartesian Coordinates for the Current Position Command of Each Axis:

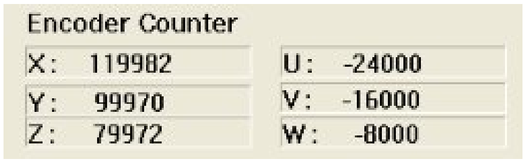


Current Position [✓ pulse] mm

X : 0	U : 0
Y : 0	V : 0
Z : 0	W : 0

MCC_GetCurPos can be used to acquire the Cartesian coordinates for the current position command of each axis.

Encoder Counter for the Current Position of Each Axis:

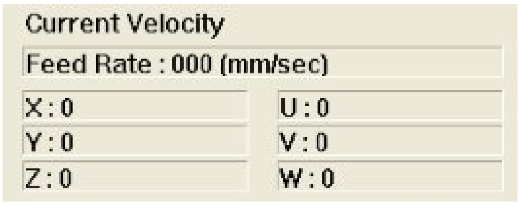


Encoder Counter

X : 119982	U : -24000
Y : 99970	V : -16000
Z : 79972	W : -8000

If an encoder is installed in the system, the encoder can use MCC_GetENCValue to acquire the encoder counter for the current position of each axis

Current Actual Speed for Each Axis and Feed Rate:



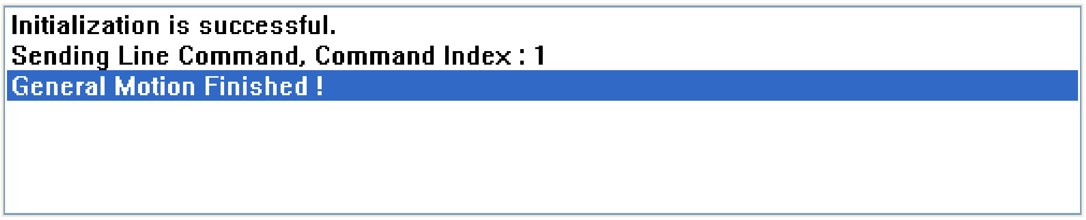
Current Velocity

Feed Rate : 000 (mm/sec)

X : 0	U : 0
Y : 0	V : 0
Z : 0	W : 0

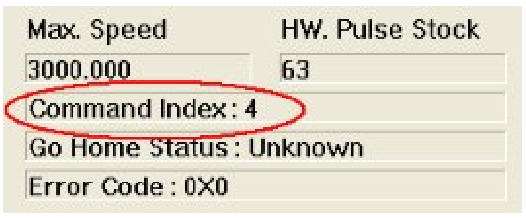
Calling MCC_GetCurFeedSpeed and MCC_GetSpeed can obtain the current actual speed at each axis and the feed rate for general motion (excluding point-to-point motion).

Information Window:



```
Initialization is successful.
Sending Line Command, Command Index : 1
General Motion Finished !
```

The "**Information Window**" displays the current status of motions and the indexes of motion commands sent to the motion command queue when the "**Run**" button in the "**Motion Command Parameter Settings**" section is clicked. These indexes can be obtained from the command return values (for example, the command return value for MCC_Line). Information about motion commands currently being executed, including motion command indexes, can be obtained using MCC_GetCurCommand. An executing motion command index is displayed below.



Max. Speed	HW. Pulse Stock
3000.000	63

Command Index : 4
Go Home Status : Unknown
Error Code : 0X0

Fig. 12 is the "**Home Sensor, Limit Switch Sensor, and Emergency Stop Status"** display area, used to display the statuses for these input points.

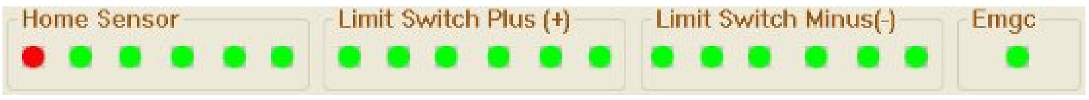


Fig. 12

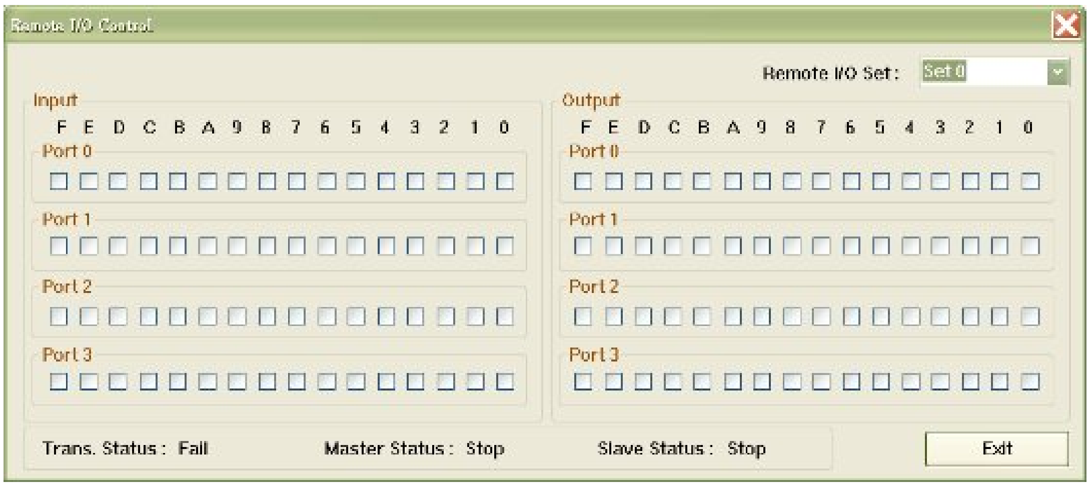
MCC_GetLimitSwitchStatus, MCC_GetGoHomeStatus, and MCC_GetEmgcStopStatus can be used to acquire the statuses of the home sensor, limit switch sensor, and emergency stop.

IX. Remote I/O Testing

If the system is installed with an remote I/O control card, the Remote I/O button can be clicked to obtain the remote I/O control window once the system is successfully initialized. Note that to use the remote I/O function normally, calling the commands listed below after the system uses MCC_InitSystem to successfully initialize the system is required.

MCC_EnableRIOSetControl();
MCC_EnableRIOSlaveControl()

Fig. 13 is the remote I/O control window.



Remote I/O Set: Set 0

Trans. Status : Fail   Master Status : Stop   Slave Status : Stop

Fig. 13

MCC_GetRIOInputValue and MCC_SetRIOOutputValue can be used to acquire and set the remote I/O information status, respectively.